EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	8	joseph near laura.in.	US-PGPUB; USPAT	OR	ON	2007/04/09 08:25
S2	772	sprint.as.	US-PGPUB; USPAT	OR	ON	2007/04/09 08:26
S3	0	S2 and (cobol and socket).clm.	US-PGPUB; USPAT	OR	ON	2007/04/09 08:26
S4	4	S2 and (cobol).clm.	US-PGPUB; USPAT	OR	ON	2007/04/09 08:26
S5	1506	717/114-119.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 08:36
S6	603	709/237.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 08:37
S7	108	(S5 or S6) and cobol	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 08:37
S8	9	S7 and socket	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 08:38
S9	346	cobol and socket	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 08:38
S10	32	S9 and (socket near communicat$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 08:39

EAST Search History

S11	50	("20050097537" "20050097539" "20050097564" "5794240" "6182276" "5649203" "5652889" "4403283" "4536840" "4866610" "4903199" "5481740" "5493675" "5613117" "5652909" "5659753" "5845121" "4937740" "4374408" "4558411" "4587628" "5225978" "5428780" "5455949" "5497500" "5555411" "5630137" "5666533" "5724564" "5737622" "5970250" "6073157" "20040060048" "6058460" "5632036" "5745768" "6090155" "20050097538" "5418965" "5404555" "6360359" "3930236" "3886523" "4005391" "4135242" "4246638" "4445177" "4447875" "4455604" "4455603").pn.	US-PGPUB; USPAT	OR	ON	2007/04/09 08:59
S12	11	("3891974"|"4320451"|"4525780"|"4636948").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 09:06
S13	2	("5146593").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 09:07
S14	2	("6453464").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 09:07
S15	96	cobol.ti.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 09:18
S16	78	S15 and (@pd<"20031030" or @ad<"20031030" or @prad<"20031030" or @rlad<"20031030")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 14:14
S17	3	("5754830" | "6621505" | "6728769").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/04/09 09:21

EAST Search History

S18	17	cobol and "socket communication"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 09:46
S19	217	copybook	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:27
S20	152	S19 and cobol	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:27
S21	49	S20 and socket	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:27
S22	42	S21 and (@pd<"20031030" or @ad<"20031030" or @prad<"20031030" or @rlad<"20031030")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:28
S23	12	copybook near3 map$5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:51
S25	10	copy$1book near3 map$5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:59
S26	38	copy$1book near3 memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 11:59

S27	1734	ebcdic	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 12:00
S28	1296	S27 and ascii	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 12:00
S29	171	ebcdic with convert$4 with ascii	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 12:01
S30	23	S29 and cobol	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 12:01
S32	26	(socket near3 full) with read$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 12:32
S33	33	copy$1book with map$4 with memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 14:30
S34	33	S33 and (@pd<"20031030" or @ad<"20031030" or @prad<"20031030" or @rlad<"20031030")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 14:14
S35	7	(US-5745748-$ or US-7007278-$ or US-6453464-$ or US-5754830-$ or US-6931623-$ or US-6886018-$ or US-6748380-$).did.	USPAT	OR	ON	2007/04/09 14:16

EAST Search History

S36	2	S35 and copy$1book	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 14:16
S37	5	copy$1book same map$4 same memory same cobol	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 14:30
S38	39	copy$1book same map$4 same memory	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/04/09 14:31

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

cobol SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Term used cobol

Found **3,344** of **199,915**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200
Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Relevance scale □▨▨■■

1 The early history of COBOL ■

Jean E. Sammet
January 1978 **ACM SIGPLAN Notices , The first ACM SIGPLAN conference on History of programming languages HOPL-I**, Volume 13 Issue 8
Publisher: ACM Press
Full text available: pdf(3.10 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper discusses the early history of COBOL, starting with the May 1959 meeting in the Pentagon which established the Short Range Committee which defined the initial version of COBOL, and continuing through the creation of COBOL 61. The paper gives a detailed description of the committee activities leading to the publication of the first official version, namely COBOL 60. The major inputs to COBOL are discussed, and there is also a description of how and why some of the technical decisi ...

2 Teaching COBOL in computer information systems programs: Problems, a proposal and an experiment ■

Yuksel Uckan
September 1986 **ACM SIGCSE Bulletin**, Volume 18 Issue 3
Publisher: ACM Press
Full text available: pdf(538.90 KB) Additional Information: full citation, abstract, index terms

The COBOL component in four-year computer information systems programs is crucial to the rest of the program, and should be handled with special care. Particularly in CIS programs modeled after ACM's recommendations for computer information systems, students who have become reasonably proficient in a high level programming language find a course on introductory COBOL rather unmotivating. Furthermore, experience has shown that COBOL is not an ideal language as the first procedural language in CIS ...

3 Software engineering for the Cobol environment ■

Michael Evans
December 1982 **Communications of the ACM**, Volume 25 Issue 12
Publisher: ACM Press
Full text available: pdf(822.13 KB) Additional Information: full citation, abstract, references, citings, index terms

In a attempt to improve the productivity of their 70 development staff, Skandinaviska Enskilda Banken has built an integrated set of manual and automatic tools for the implementation of Cobol programs. It was possible to use a number of modern programming techniques, including software engineering methods, in a Cobol environment. The project required 31 person-months; the aims, current status, and initial results are reported.

Keywords: Cobol tools, business programming, commercial programming

4 Cobol vs. PL/1: some performance comparisons

Paul J. Jalics

March 1984 **Communications of the ACM**, Volume 27 Issue 3

Publisher: ACM Press

Full text available: pdf(445.59 KB) Additional Information: full citation, references, index terms

5 "The Cobol dinosaur": does it face extinction?

Malini Krishnamurthi

September 1998 **ACM SIGUCCS Newsletter**, Volume 28 Issue 1-3

Publisher: ACM Press

Full text available: pdf(525.73 KB) Additional Information: full citation, abstract, references, index terms

COBOL (common business oriented language) is a traditional third generation language that was introduced in 1959 and has been in use from then until now. Its longevity seems awesome to many.

6 COBOL on a PC: a new perspective on a language and its performance

Paul J. Jalics

February 1987 **Communications of the ACM**, Volume 30 Issue 2

Publisher: ACM Press

Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A comparison of Cobol performance on the PC AT Enhanced versus an IBM 370 mainframe suggests that high-quality PC compiler implementations—combined with the new language features of the Cobol 85 Standard—are improving the PC environment for Cobol to the point where serious applications can now be developed and debugged on the PC, either to be run on the PC itself, or for eventual uploading to a mainframe.

7 What does aspect-oriented programming mean to Cobol?

Ralf Lämmel, Kris De Schutter

March 2005 **Proceedings of the 4th international conference on Aspect-oriented software development AOSD '05**

Publisher: ACM Press

Full text available: pdf(172.28 KB) Additional Information: full citation, abstract, references, citings, index terms

We study AOP in the context of business programming with Cobol. We face the following questions: What are join points in Cobol programs? What is advice? Does classic Cobol provide any constructs that hint at AOP? (Yes!) What are typical crosscutting concerns in the Cobol world? How do otherwise typical crosscutting concerns make sense for Cobol? How does AOP for Cobol align with classic re-engineering transformations for Cobol? We deliver an AOP language design for Cobol. Codename: AspectCobol. ...

Keywords: Cobol, aspect-oriented programming, business programming

8 In defense of teaching structured COBOL as computer science: (or, notes on being sage struck)

Howard E. Tompkins

April 1983 **ACM SIGPLAN Notices**, Volume 18 Issue 4

Publisher: ACM Press

Full text available: pdf(824.59 KB) Additional Information: full citation, references, citings

9 Advanced structured COBOL programming
Asad Khailany
February 1977 **ACM SIGCSE Bulletin , Proceedings of the seventh SIGCSE technical symposium on Computer science education SIGCSE '77**, Volume 9 Issue 1
Publisher: ACM Press
Full text available: pdf(306.76 KB) Additional Information: full citation, abstract, references, citings, index terms

There is a large market demand for COBOL programmers. Can a single introductory course in COBOL programming provide the students with sufficient professional expertise to qualify them for these job openings? If not, is it feasible to offer a second course in COBOL programming? In this paper the technical capabilities of those who have completed an introductory COBOL programming course and the skills and technical qualifications which are desired by employers of entry level COBOL programmers ...

10 Comparative performance of COBOL programs on mini vs. large computer systems
Paul J. Jalics
August 1978 **ACM SIGMINI Newsletter , Proceedings of the first SIGMINI symposium on Small systems SIGMINI '78**, Volume 4 Issue 4
Publisher: ACM Press
Full text available: pdf(574.64 KB) Additional Information: full citation, abstract, references, index terms

The comparative performance characteristics of COBOL programs in a small versus large computer systems are investigated. The vehicle consists of a set of synthetic benchmark COBOL programs, each measuring a particular aspects of COBOL programs; and in addition a large actual COBOL program. Measurement of the CPU execution time and the elapsed clock time for various COBOL computations, data manipulation, and input/output is made on both a large scale computer (IBM 370/158) and a minicomputer ...

11 Predicting potential COBOL performance on low level machine architectures
Jerome A. Otto
October 1985 **ACM SIGPLAN Notices**, Volume 20 Issue 10
Publisher: ACM Press
Full text available: pdf(926.87 KB) Additional Information: full citation, abstract, index terms

As a COBOL host, a computer architecture should efficiently execute those language constructs that are most frequently used in actual programs. However, when the language's control and data structures are at a far higher level than the control and data structures of the underlying machine, the compiler designer is faced with a large number of potential choices for mapping these high level structures to the low level architecture. This is the case for implementing COBOL on a typical mini-computer ...

12 Facilitating COBOL programmers' transition to the C language
Ritu Agarwal, Jayesh Prasad
April 1997 **Proceedings of the 1997 ACM SIGCPR conference on Computer personnel research SIGCPR '97**
Publisher: ACM Press
Full text available: pdf(1.27 MB) Additional Information: full citation, references, citings, index terms

13 Teaching personal computer Cobol with Watcom Cobol
Joseph W. Jr. Trigg
February 1987 **ACM SIGCSE Bulletin , Proceedings of the eighteenth SIGCSE technical symposium on Computer science education SIGCSE '87**, Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(350.31 KB) Additional Information: full citation, abstract, references, index terms

The teaching transition from a mainframe language to a subset of the same language for personal computer use could be an unpleasant experience without adequate preparation. This paper is intended to be of benefit for the teachers of COBOL who are contemplating a change from mainframe COBOL to the WATCOM COBOL interpreter.

14 Experience with porting techniques on a COBOL 74 compiler
I. M. Kipps
June 1982 **ACM SIGPLAN Notices , Proceedings of the 1982 SIGPLAN symposium on Compiler construction SIGPLAN '82**, Volume 17 Issue 6
Publisher: ACM Press
Full text available: pdf(582.06 KB) Additional Information: full citation, abstract, references, index terms

The problems of compiler construction have largely been solved for COBOL '74, but a remaining fundamental consideration in a commercial environment is the cost of compiler development. This can be reduced by the use of portable software, but the cost of porting to a new system remains significant. In the microprocessor system market, two approaches have been followed. The first is to produce software for a virtual machine environment and simulate this on real hardware be means of an interpr ...

15 A study of errors, error-proneness, and error diagnosis in Cobol
Charles R. Litecky, Gordon B. Davis
January 1976 **Communications of the ACM**, Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(517.12 KB) Additional Information: full citation, abstract, references, citings

This paper provides data on Cobol error frequency for correction of errors in student-oriented compilers, improvement of teaching, and changes in programming language. Cobol was studied because of economic importance, widespread usage, possible error-inducing design, and lack of research. The types of errors were identified in a pilot study; then, using the 132 error types found, 1,777 errors were classified in 1,400 runs of 73 Cobol students. Error density was high: 20 percent of the types ...

Keywords: Cobol, diagnostics, error analysis, error correction, error frequency, error-proneness, errors in programming, learning of programming, programming language errors, spelling errors, syntactic errors, teaching of programming

16 MCOBOL—a prototype macro facility for Cobol
J. M. Triance, J. F.S. Yow
August 1980 **Communications of the ACM**, Volume 23 Issue 8
Publisher: ACM Press
Full text available: pdf(2.12 MB) Additional Information: full citation, references, citings

Keywords: Cobol, extensibility, language, macros, portability, syntax macros

17 Re-engineering legacy Cobol programs
J. K. Joiner, W. T. Tsai
May 1998 **Communications of the ACM**
Publisher: ACM Press
Full text available: pdf(116.24 KB) Additional Information: full citation, references, index terms

18 Why COBOL programmers refuse Ada
Kenneth Fussichen
December 1990 **Proceedings of the conference on TRI-ADA '90 TRI-Ada '90**
Publisher: ACM Press

Full text available: pdf(696.76 KB) Additional Information: full citation, abstract, references

The problems associated with asking a COBOL1 programmer to learn Ada can be likened to the problems you might anticipate when an American employee is asked to learn Japanese. Significant motivation must be provided to overcome some of the obstacles. First, learning Japanese is difficult. The Japanese and English languages have no common ancestry. Second, societal values vary. Third, it is difficult to generate much interest in learning anything when its anticipated prac ...

19 Can visual basic replace COBOL? ...and should it?

Deborah L. Dunn, Dennis Lingerfelt
April 2005 **Journal of Computing Sciences in Colleges**, Volume 20 Issue 4
Publisher: Consortium for Computing Sciences in Colleges
Full text available: pdf(137.96 KB) Additional Information: full citation, abstract, references, index terms

Computer science educators are in a rather unusual position of being constantly forced to examine and update their curriculum to accommodate the perpetual changes in technology. This trend is especially apparent in the field of programming languages. We are regularly (almost too often it sometimes seems) asking ourselves "What language is the best for <fill in the blank> applications?" As educators, we are often hesitant to "jump on the bandwagon" quite frankly because often the bandwagon ...

20 Structured programming in Cobol: an approach for application programmers

Allen van Gelder
January 1977 **Communications of the ACM**, Volume 20 Issue 1
Publisher: ACM Press
Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings

Techniques for designing and writing Cobol programs are presented. Previous work in structured programming is drawn upon and adapted. The presentation is informal: the terminology is nonmathematical as far as possible, no theorems are proved, and examples are used frequently. Top-down program design is implemented through the use of structured flowcharts, disciplined specifications, and step by step verification. A well-formed Cobol program is defined. The proper use of the GO TO and other ...

Keywords: Cobol, GO TO statement, application programming, flowchart, program verification, repeat statement, software reliability, structured programming, top-down, well-formed program

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next